


SOFTWARE

Open Access



# CAMISIM: simulating metagenomes and microbial communities

Adrian Fritz<sup>1†</sup>, Peter Hofmann<sup>1,2†</sup>, Stephan Majda<sup>1,2</sup>, Eik Dahms<sup>1,2</sup>, Johannes Dröge<sup>1,2</sup>,  
Jessika Fiedler<sup>1,2</sup>, Till R. Lesker<sup>1,3</sup>, Peter Belmann<sup>1,4</sup>, Matthew Z. DeMaere<sup>5</sup>, Aaron E. Darling<sup>5</sup>,  
Alexander Sczyrba<sup>4</sup>, Andreas Bremges<sup>1,3</sup> and Alice C. McHardy<sup>1,2\*</sup> 

## Abstract

**Background:** Shotgun metagenome data sets of microbial communities are highly diverse, not only due to the natural variation of the underlying biological systems, but also due to differences in laboratory protocols, replicate numbers, and sequencing technologies. Accordingly, to effectively assess the performance of metagenomic analysis software, a wide range of benchmark data sets are required.

**Results:** We describe the CAMISIM microbial community and metagenome simulator. The software can model different microbial abundance profiles, multi-sample time series, and differential abundance studies, includes real and simulated strain-level diversity, and generates second- and third-generation sequencing data from taxonomic profiles or de novo. Gold standards are created for sequence assembly, genome binning, taxonomic binning, and taxonomic profiling. CAMSIM generated the benchmark data sets of the first CAMI challenge. For two simulated multi-sample data sets of the human and mouse gut microbiomes, we observed high functional congruence to the real data. As further applications, we investigated the effect of varying evolutionary genome divergence, sequencing depth, and read error profiles on two popular metagenome assemblers, MEGAHIT, and metaSPAdes, on several thousand small data sets generated with CAMISIM.

**Conclusions:** CAMISIM can simulate a wide variety of microbial communities and metagenome data sets together with standards of truth for method evaluation. All data sets and the software are freely available at <https://github.com/CAMI-challenge/CAMISIM>

**Keywords:** Metagenomics software, Microbial community, Benchmarking, Simulation, Metagenome assembly, Genome binning, Taxonomic binning, Taxonomic profiling, CAMI

## Introduction

Extensive 16S rRNA gene amplicon and shotgun metagenome sequencing efforts have been and are being undertaken to catalogue the human microbiome in health and disease [1, 2] and to study microbial communities of medical, pharmaceutical, or biotechnological relevance [3–8]. We have since learned that naturally occurring microbial communities cover a wide range of organisational complexities—with populations ranging from half

a dozen to likely tens of thousands of members—can include substantial strain level diversity and vary widely in represented taxa [9–12]. Analyzing these diverse communities is challenging.

The problem is exacerbated by use of a wide range of experimental setups in data generation and the rapid evolution of short- and long-read sequencing technologies [13, 14]. Owing to the large diversity of generated data, the possibility to generate realistic benchmark data sets for particular experimental setups is essential for assessing computational metagenomics software.

CAMI, the initiative for the Critical Assessment of Metagenome Interpretation, is a community effort aiming to generate extensive, objective performance overviews of

\*Correspondence: [alice.mchardy@helmholtz-hzi.de](mailto:alice.mchardy@helmholtz-hzi.de)

<sup>†</sup>Adrian Fritz and Peter Hofmann contributed equally to this work.

<sup>1</sup>Computational Biology of Infection Research, Helmholtz Centre for Infection Research, 38124 Braunschweig, Germany

<sup>2</sup>Formerly Department of Algorithmic Bioinformatics, Heinrich-Heine University Düsseldorf, 40225 Düsseldorf, Germany

Full list of author information is available at the end of the article



computational metagenomics software [15]. CAMI organizes benchmarking challenges and encourages the development of standards and reproducibility in all aspects, such as data generation, software application, and result interpretation [16].

We here describe CAMISIM, which was originally written to generate the simulated metagenome data sets used in the first CAMI challenge. It has since been extended into a versatile and highly modular metagenome simulator. We demonstrate the usability and utility of CAMISIM with several applications. We generated complex, multi-replicate benchmark data sets from taxonomic profiles of human and mouse gut microbiomes [1, 17]. We also simulated thousands of small “minimally challenging metagenomes” to characterize the effect of varying sequencing coverage, evolutionary divergence of genomes, and sequencing error profiles on the popular MEGAHIT [18] and metaSPAdes [19] assemblers.

### The CAMISIM software

CAMISIM allows customization of many properties of the generated communities and data sets, such as the overall number of genomes (community complexity), strain diversity, the community genome abundance distributions, sample sizes, the number of replicates, and sequencing technology used. For setting these options, a configuration file is needed, which is described in Additional file 1. Simulation with CAMISIM has three stages (Fig. 1):

- 1 Design of the community, which includes selection of the community members and their genomes, and assigning them relative abundances,
- 2 Metagenome sequencing data simulation, and
- 3 Postprocessing, where the binning and assembly gold standards are produced.

### Community design

In this step, the community genome abundance profiles, called  $P_{out}$ , are created. These also represent the gold

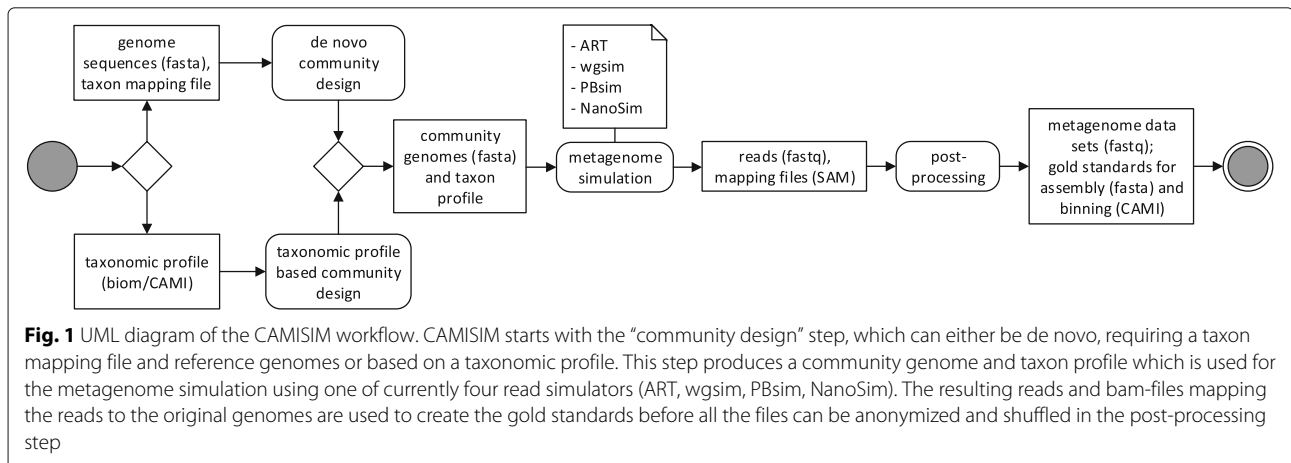
standard for taxonomic profiling and, from the strain to the superkingdom rank, specify the relative abundances of individual strains (genomes) or their parental taxa in percent. In addition, a genome sequence collection for the strains in  $P_{out}$  is generated. Both  $P_{out}$  and the genome sequence collection are needed for the metagenome simulation in step 2. The taxonomic composition of the simulated microbial community is either determined by user-specified taxonomic profiles or generated de novo by sampling from available genome sequences.

### Profile-based design

Taxonomic profiles can be provided in BIOM (Biological Observation Matrix) format [20]. With input profiles, the NCBI complete genomes [21] are used as the sequence collection for creating metagenome data sets. Optionally, the user can choose to also include genomes marked as “scaffold” or “contig” by the NCBI. Input genomes are split at positions with multiple occurrences of ambiguous bases, such that no reads spanning contig borders within larger scaffolds are simulated.

Profiles can include bacterial, archaeal, and eukaryotic taxa, as well as viruses. The taxonomic identifiers of BIOM format are interpreted as free text scientific names and are mapped to NCBI taxon IDs (algorithm in Additional file 1). The so generated input profile  $P_{in}$  specifies pairs  $(t, ab_t)$  of taxon IDs  $t$  and taxon abundances  $ab_t \in \mathbb{R}_{\geq 0}$ . The profile taxa are usually defined at higher ranks than strain and thus have to be mapped approximately to the genome sequence collection for creating  $P_{out}$ .

Given an ordered list of ranks  $R = (species, genus, family, order, class, phylum, superkingdom)$ , CAMISIM requires as an additional parameter a highest rank  $r_{max} \in R$ . We define the binary operator  $<$  based on the ordering of the ranks in  $R$ . Given two ranks,  $r_i, r_j \in R$ , we write  $r_i < r_j$ , if  $r_i$  appears before  $r_j$  in  $R$ , and we say  $r_i$  is below  $r_j$ . Related complete genomes are searched for all ranks below  $r_{max}$ .



By default, this is the *family* rank. Another parameter is the maximum number of strains  $m$  that are included for an input taxon in a simulated sample.

To create  $P_{out}$  from  $P_{in}$ , the following steps are performed: let  $G_{in}$  be the set of taxon IDs of the genome collection at the lowest annotated taxonomic rank, usually *species* or *strain*. For all  $t \in G_{in}$ , the reference taxonomy specifies a taxonomic lineage of taxon IDs (or undefined values) across the considered ranks in  $R$ . We use these to identify a collection of sets  $F = \{G_t \mid t = \text{lineage taxon represented by } \geq 1 \text{ complete genome}\}$ , which specifies for each lineage taxon the taxon IDs of available genomes from the genome collection.  $F$  is used as input for Algorithm 1.

---

**Algorithm 1:** Creating a community genome abundance profile; *genome-select* ( $F, P_{in}, m, r_{max}$ )

---

**input** : Collection of sets  $F$  of taxonomic IDs of available complete genomes, taxonomic profile  $P_{in}$ , maximum strains per OTU  $m$ , highest rank  $r_{max}$  considered for similarity

**output:** Community genome abundance profile  $P_{out}$

```

1  $P_{out} = \emptyset$ 
2 foreach  $(t, ab_t) \in P_{in}$  do
3   get lineage path  $tax_t$  from reference taxonomy
4   foreach rank  $r \in R < r_{max}$  do
5      $t_r = tax_t$  on rank  $r$ ; // check whether
    a complete genome for taxon  $t_r$ 
    exists
6     if  $t_r \in F$  then
7        $G_{t_r}$  = set of available full genomes
    corresponding to taxon  $t_r$  in  $F$ 
8       draw a random number  $X$  from truncated
    geometric distribution (Eq. 1)
9       if  $X < |G_{t_r}|$  then
10         $G_{selected}$  = randomly select  $X$ 
    genomes from  $G_{t_r}$ 
11      else
12         $G_{selected} = G_{t_r}$ 
13         $Y$  = list of  $|G_{selected}|$  random numbers
    from lognormal distribution (Eq. 2)
14        foreach  $i \in G_{selected}$  do
15           $ab_i = \frac{Y_i}{\sum_{i \in G_{selected}} Y_i} \cdot ab_t$  (Eq. 3)
16          add  $(i, ab_i)$  to  $P_{out}$ 
17          remove  $i$  from  $G_{t_r}$ 
18        break; // if a complete genome
    exists, continue with the
    next taxon instead of rank
19      else
20        issue "Unmapped genome" warning
21 return  $P_{out}$ 

```

---

The algorithm retrieves for each  $t$  from the tuples  $(t, ab_t) \in P_{in}$  the lineage path  $tax_t$  across the ranks of  $R$  (lines 2–3). Moving from the species to the highest considered rank,  $r_{max}$ , the algorithm determines whether for a lineage taxon  $t_r$  at the considered rank  $r$  a complete genome exists, that is, whether  $G_t \neq \emptyset$  for  $t = t_r$  (lines 4–5). If this is the case, the search ends and  $t_r$  is considered further (line 6). If no complete genome is found for a particular lineage, the lineage is not included in the simulated community, and a warning is issued (line 20). Next, the number of genomes  $X$  with their taxonomic IDs  $t_r$  to be added to  $P_{out}$  is drawn from a *truncated geometric distribution* (Eq. 1, line 8) with a mean of  $\mu = \frac{m}{2}$  and the parameter  $k$  restricted to be less than  $m$ .

$$P(X = k) = \left(1 - \frac{1}{\mu}\right)^k \cdot \frac{1}{\mu} \tag{1}$$

If  $|G_{t_r}|$  is less than  $X$ ,  $G_{t_r}$  is used entirely as  $G_{selected}$ , the genomes of  $t_r$  that are to be included in the community. Otherwise  $X$  genomes are drawn randomly from  $G_{t_r}$  to generate  $G_{selected}$  (lines 9–12). It is optional to use genomes multiple times, by default the selected genomes  $g \in G_{selected}$  are removed from  $F$ , such that no genome is selected twice (line 17). Based on the taxon abundances  $ab_t$  from  $P_{in}$ , the abundances  $ab_i$  of the selected taxa  $i \in G_{selected}$  for  $t$  are then inferred. First, random variables  $Y_i$  are drawn from a configurable lognormal distribution, with by default normal mean  $\mu = 1$  and normal standard deviation  $\sigma = 2$  (Eq. 2), and then the  $ab_i$  are set (Eq. 3; lines 13–15). Finally, the created pairs  $(i, ab_i)$  are added to  $P_{out}$  (line 16) and  $P_{out}$  is returned (line 21).

$$Y_i \sim \text{Lognormal}(\mu, \sigma)$$

$$\Leftrightarrow \frac{d}{dx} P(Y_i \leq x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}} \tag{2}$$

$$ab_i = \frac{Y_i}{\sum_{j \in G_{selected}} Y_j} \cdot ab_t \tag{3}$$

### De novo design

A genome sequence collection to sample and a mapping file have to be specified. The mapping file defines for each genome a taxonomic ID (per default from the NCBI taxonomy), a novelty category and an operational taxonomic unit (OTU) ID. Grouping genomes into OTUs is required for sampling related genomes, to increase strain-level diversity in the simulated microbial communities. The novelty category reflects how closely a query genome is related to draft or complete genomes in a genome sequence reference collection. This is used to maximize the spread of selected genomes across the range of taxonomic distances to the genome reference collection, such that there are genomes included of "novel" strains,

species, or genera. This distinction is relevant for evaluating reference-based taxonomic binners and profilers, which may perform differently across these different categories. The user can manually generate the mapping file as described in Additional file 1 or in [15].

If controlled sampling of strains is not required, every genome can be assigned to a different OTU ID. If no reference-based taxonomic binners or profilers are to be evaluated, or the provided genome sequence collection does not vary much in terms of taxonomic distance to publicly available genomes used as references for these programs, all genomes can be assigned the same novelty category.

In addition, the number of genomes  $g_{\text{real}}$  to be drawn from the input genome selection and the total number of genomes  $g_{\text{tot}}$  for the community genome abundance profile  $P_{\text{out}}$  have to be specified. The  $g_{\text{real}}$  real genomes are drawn from the provided genome sampling collection. An equal number of genomes is drawn for every novelty category. If the number of genomes for a category is insufficient, proportionately more are drawn from others. In addition, CAMISIM simulates  $g_{\text{sim}} = g_{\text{tot}} - g_{\text{real}}$  genomes of closely related strains from the chosen real genomes in total. These genomes are created with an enhanced version of sgEvolver [22] (Additional file 1: Methods) from a subset of randomly selected real genomes. Given  $m$ , the maximum number of strains per OTU, up to  $m - 1$  simulated strain genomes are added *per genome*. The exact number of genomes  $X$  to be simulated for a selected OTU is drawn from a geometric distribution with mean  $\mu = 0.3^{-1}$  (Eq. 1). This procedure is repeated until  $g_{\text{sim}}$ -related genomes have been added to the community genome collection, comprising  $g_{\text{tot}} = g_{\text{real}} + g_{\text{sim}}$  genomes [15].

Next, community genomes are assigned abundances. The relevant user-defined parameters for this step are the sample type and the number of samples  $n$ . In addition to single samples, multi-sample data sets (with differential abundances, replicates or time series) have become widely used in real sequencing studies [23–26], also due to their utility for genome recovery using covariance-based genome binners such as CONCOCT [27] or MetaBAT [28]. Several options for creating multi-sample metagenome data sets with these setups are provided:

- 1 If simulating a *single sample data set*, the relative abundances are drawn from a lognormal distribution, which is commonly used to model microbial communities [29–32]. The two parameters of the lognormal distribution can be changed. By default, the mean is set to 1 and the standard deviation to 2 (Eq. 2). Setting the standard deviation  $\sigma$  to 0 results in a uniform distribution.
- 2 The *differential abundance mode* models a community sampled multiple times after the

environmental conditions or the DNA extraction protocols (and accordingly the community abundance profile) have been altered. This mode creates  $n$  different lognormally (Eq. 2) distributed genome abundance profiles.

- 3 Metagenome data sets with multiple samples with very similar genome abundance distributions can be created using the *replicates mode*. Having multiple replicates of the same metagenome has been reported to improve the quality for some metagenome analysis software, such as for genome binners [23, 27, 33, 34]. Based on an initial log-normal distribution  $D_0$ ,  $n$  samples are created by adding Gaussian noise to this initial distribution (Eq. 4). The Gaussian term accounts for all kinds of effects on the genome abundances of the metagenomic replicates including, but not limited to, different experimenters, different place of extraction, or other batch effects.

$$D_i = D_0 + \varepsilon \text{ with } \varepsilon \sim N(0, 1) \text{ and } \varepsilon \sim N(0, 1) \tag{4}$$

$$\Leftrightarrow \frac{d}{dx} P(\varepsilon \leq x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2}$$

- 4 *Time series* metagenome data sets with multiple related samples can be created. For these, a Markov model-like simulation is performed, with the distribution of each of the  $n$  samples (Eq. 5) depending on the distribution of the previous sample plus an additional either lognormal (Eq. 2) or Gaussian (Eq. 4) term. This emulates the natural process of fluctuating abundances over time and ensures that the abundance changes to the previously sampled metagenome do not grow very large.

$$D_i = D_{i-1} + \varepsilon \quad \text{with } D_0 \sim \text{Lognormal}(\mu, \sigma) \quad \text{and } \varepsilon \sim N(0, 1) \quad \text{or} \tag{5}$$

$$D_i = \frac{D_{i-1} + \varepsilon}{2} \quad \text{with } \varepsilon \sim \text{Lognormal}(\mu, \sigma)$$

### Metagenome simulation

Metagenome data sets are generated from the genome abundance profiles of the community design step. For each genome-specific taxon  $t$  and its abundance  $(t, ab_t) \in P_{\text{out}}$ , its genome size  $s_t$ , together with the total number of reads  $n$  in the sample, determines the number of generated reads  $n_t$  (Eq. 6). The total number of reads  $n$  is the overall sequence sample size divided by the mean read-length of the utilized sequencing technology.

$$n_t = n \cdot \frac{ab_t \cdot s_t}{\sum_{i \in P_{\text{out}}} ab_i \cdot s_i} \tag{6}$$

By default, ART [35] is used to create Illumina 2 × 150 bp paired-end reads with a HiSeq 2500 error profile. The profile has been trained on MBarcode-26 [36], a defined mock community that has already been used to benchmark bioinformatics software and a full-length 16S rRNA gene amplicon sequencing protocol [37, 38], and is distributed with CAMISIM. Other ART profiles, such as the one used for the first CAMI challenge, can also be used. Further available read simulators are wgsim (<https://github.com/lh3/wgsim>, originally part of SAMtools [39]) for simulating error-free short reads, pbsim [40] for simulating Pacific Biosciences data and nanosim [41] for simulating Oxford Nanopore Technologies reads. The read lengths and insert sizes can be varied for some simulators.

For every sample of a data set, CAMISIM generates FASTQ files and a BAM file [39]. The BAM file specifies the alignment of the simulated reads to the reference genomes.

### Gold standard creation and postprocessing

From the simulated metagenome data sets—the FASTQ and BAM files—CAMISIM creates the assembly and binning gold standards. The software extracts the perfect assembly for each individual sample, and a perfect co-assembly of all samples together by identifying all genomic regions with a coverage of at least one using SAMtools' mpileup and extracting these as error-free contigs. This gold standard does not include all genome sequences available for the simulation, but the best possible assembly of their sampled reads.

CAMISIM generates the genome and taxon binning gold standards for the reads and assembled contigs, respectively. These specify the genome and taxonomic lineage that the individual sequences belong to. All sequences can be anonymized and shuffled (but tracked throughout the process), to enable their use in benchmarking challenges. Lastly, files are compressed with gzip and written to the specified output location.

## Results

### Comparison to the state-of-the-art

We tested seven simulators and compared them to CAMISIM (Table 1). All generate Illumina data and some—NeSSM [42], BEAR [43], FASTQSim [44], and Grinder [45]—also use a taxonomic profile. Novel and unique to CAMISIM is the ability to simulate long-read data from Oxford Nanopore, of hybrid data sets with multiple sequencing technologies and multi-sample data sets, such as with replicates, time series, or differential abundances. Grinder [45] can also create multiple samples, but only with differential abundances. In addition, CAMISIM creates gold standards for assembly (single sample assemblies and multi-sample co-assemblies), for taxonomic and genome binning of reads or contigs and for taxonomic profiling. Finally, CAMISIM can evolve multiple strains for selected input genomes and allows specification of the degree of real and simulated intra-species heterogeneity within a data set.

### Effect of data properties on assemblies

We created several thousand “minimally challenging” metagenome samples by varying one data property relevant for assembly, while keeping all others the same. Using these, we studied the effect of evolutionary divergence between genomes, different error profiles, and coverage on the popular metaSPAdes [19], version 3.12.0, and MEGAHIT [18], versions 1.1.2 and 1.0.3, assemblers, to systematically investigate reported performance declines for assemblers in the presence of strain-level diversity, uneven coverage distributions, and abnormal error profiles [15, 46, 47]. Both MEGAHIT and metaSPAdes work on de Bruijn graphs, which are created by splitting the input reads into smaller parts, the  $k$ -mers, and connecting two  $k$ -mers if they overlap by exactly  $k-1$  letters. For every sequencing error  $k$  erroneous  $k$ -mers are introduced into the de Bruijn graph, which might substantially impact assembly (Fig. 2).

**Table 1** Properties of popular metagenome sequence simulators

Software	De novo	Profile	Multi	Strains	Non-Illumina data	Licensed	Updated
MetaSim [62]	✓	X	X	✓	454	P, AU	03/2009
iMESS [63]	✓	X	X	X	454	–	07/2014
BBMap [64]	✓	X	X	X	–	LBL	01/2019
NeSSM [42]	✓	✓	X	X	454	AU	07/2013
BEAR [43]	✓	✓	X	X	–	AU	02/2017
FASTQSim [44]	✓	✓	X	X	SOLiD, IonTorrent, PacBio	GPL	05/2015
Grinder [45]	✓	✓	✓	X	Sanger, 454	GPL	04/2016
CAMISIM	✓	✓	✓	✓	PacBio, ONT, ...	Apache 2.0	01/2019

Abbreviations: P, proprietary software; AU, academic use only; LBL, Lawrence Berkeley Lab

The table shows if an abundance profile can be generated by the simulator de novo and if an existing *profile* of a microbial community can be used as input. Further inspected features are the ability to simulate *multi-sample* data sets, *strains*, and *non-Illumina data* (e.g., long reads). Lastly, the table states if and how a software is *licensed*, and the date it was last recently *updated*

### Varying genome coverage and sequencing error rates

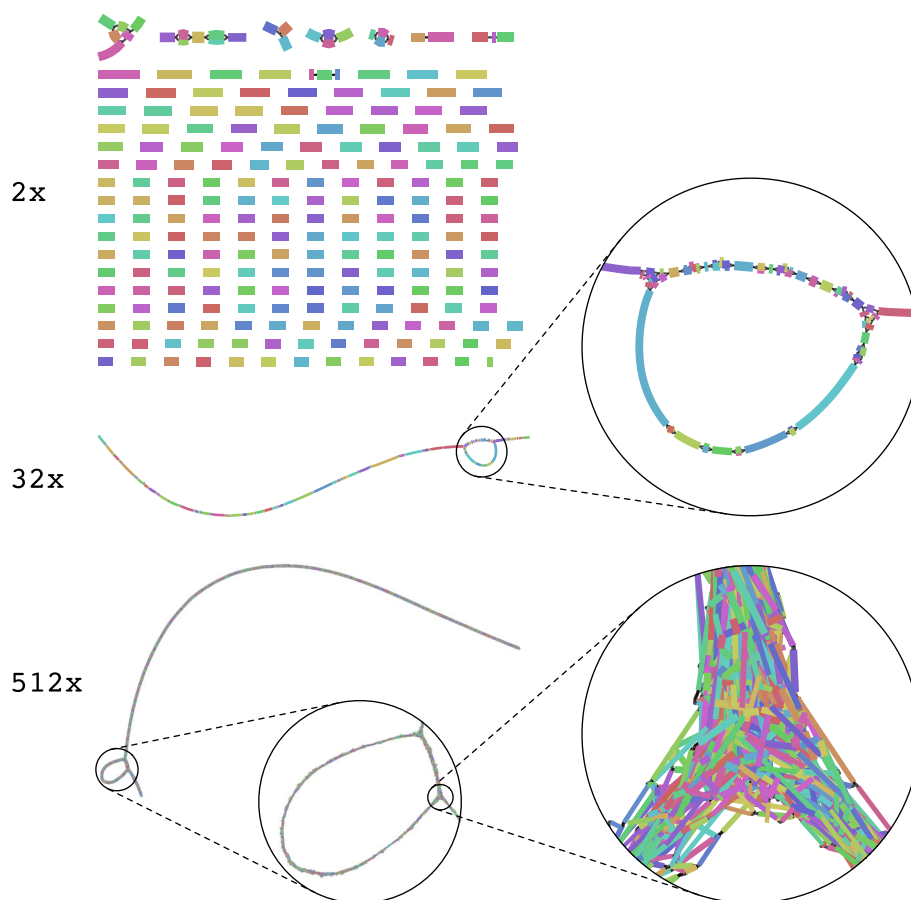
We initially simulated samples from *Escherichia coli* K12 MG1655 with varying coverage and different error rates. Reads were generated at  $512\times$  genome coverage and subsampled stepwise by 50% until  $2\times$  coverage was reached, resulting in a sample series with 512, 256, 128, 64, 32, 16, 8, 4, and 2-fold coverage, respectively. Subsampling was employed to control variation in the sampling of different genomic regions. To assess the effect of sequencing errors, four read data sets were simulated: three using wgsim with uniform error rates of 0%, 2%, and 5%, and one using ART with the CAMI challenge error profile (ART CAMI).

Both assemblers were run on these data sets with default options, except for the phred-offset parameter for metaSPAdes, which was set to 33. Both assemblers performed similar across all error rates and coverages, with assembly quality varying substantially with coverage (Fig. 3). Performance on the data generated with the 5% error profile was worst throughout. This is an unrealistically high

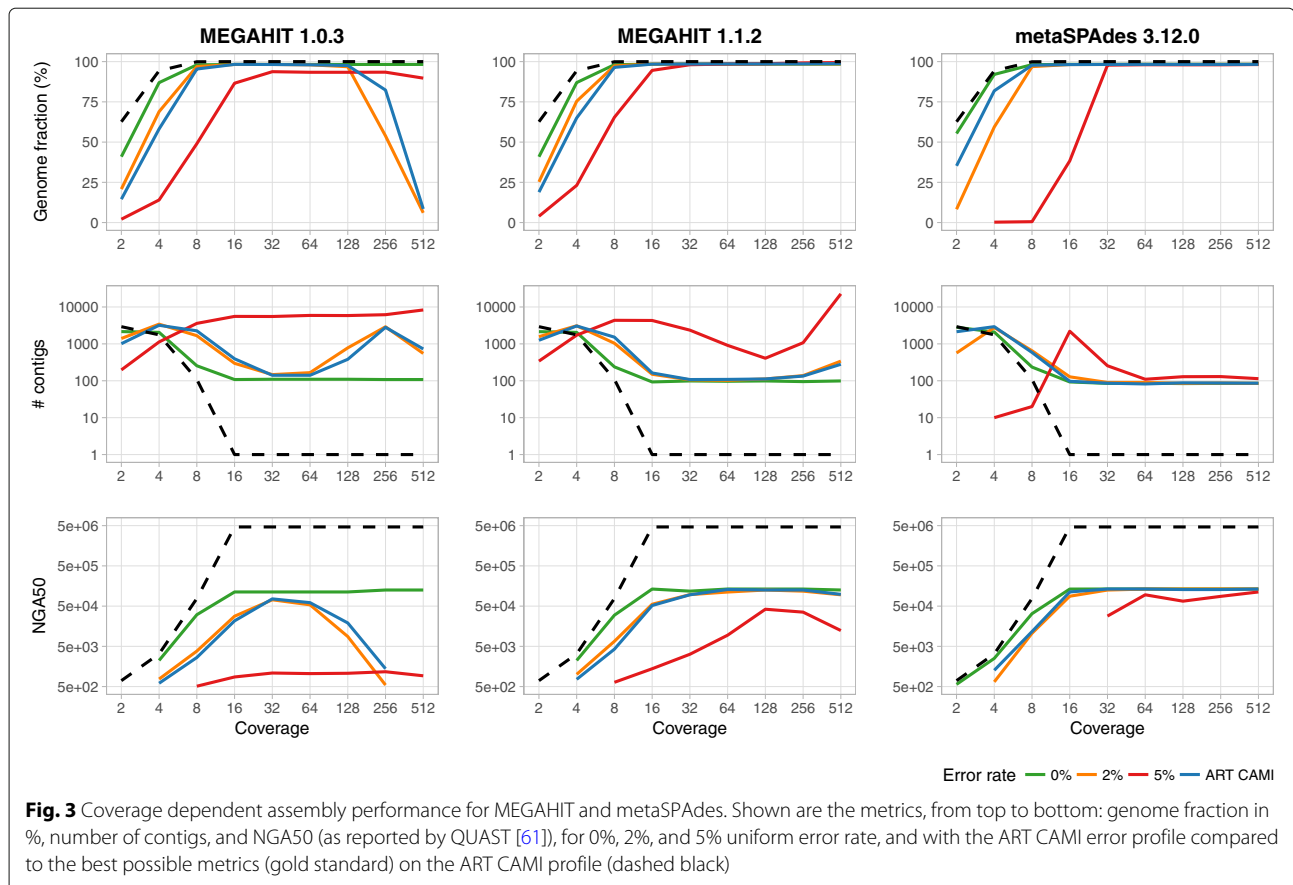
error profile for Illumina data [47] that software need not necessarily be adapted to handle well.

If coverage was low, assembly failed, generating a large number of small (low NGA50) contigs covering only a small genome portion (genome fraction) across all data sets, because of uncovered regions in the genomes. Sequencing errors (denoted  $\varepsilon$ ) do not play a major role (Fig. 2). The expected per-base error-rate  $E_p = \text{cov} \cdot \varepsilon$  (disregarding the biased errors in the short-read sequencing technologies) is far below 1 ( $E_p \ll 1$ ). With increasing coverage, assembly improved consistently across the 0%, 2%, and CAMISIM ART error profile data sets and both assemblers for all metrics (Fig. 3), and reaching an early plateau by 8–16 $\times$  coverage.

Notably, the performance of an earlier version of MEGAHIT (1.0.3) decreased substantially (declining genome fraction and NGA50) for more than 128 $\times$  coverage, except for error-free reads. For instance, at 5% error rate, MEGAHIT, version 1.0.3, generated an exponential



**Fig. 2** Assembly graphs become more complex as coverage increases. MEGAHIT assembly graphs ( $k = 41$ ) of an *E. coli* K12 genome for 2 $\times$ , 32 $\times$ , and 512 $\times$  per-base coverage, respectively, visualized with Bandage [60]. For 2 $\times$  coverage, the graph is disconnected and thus the assembly fragmented. With increasing coverage more and more unitigs can be joined, first resulting in a decent assembly for 32 $\times$  coverage, but—due to sequencing errors adding erroneous edges to the graph—a fragmented assembly again for 512 $\times$  coverage



number of contigs at high coverages, which keeps the genome fraction artificially high. For these high coverages and error rates, we expect multiple errors at every position of the genome ( $E_p \gg 10$ ). This creates de Bruijn graphs with many junctions and bubbles (Fig. 2) which cannot easily be resolved and may lead to breaking the assembly apart and covering the same part of the genome with multiple, short, and erroneous contigs (Fig. 3).

#### Effect of evolutionary relatedness on assembler performances

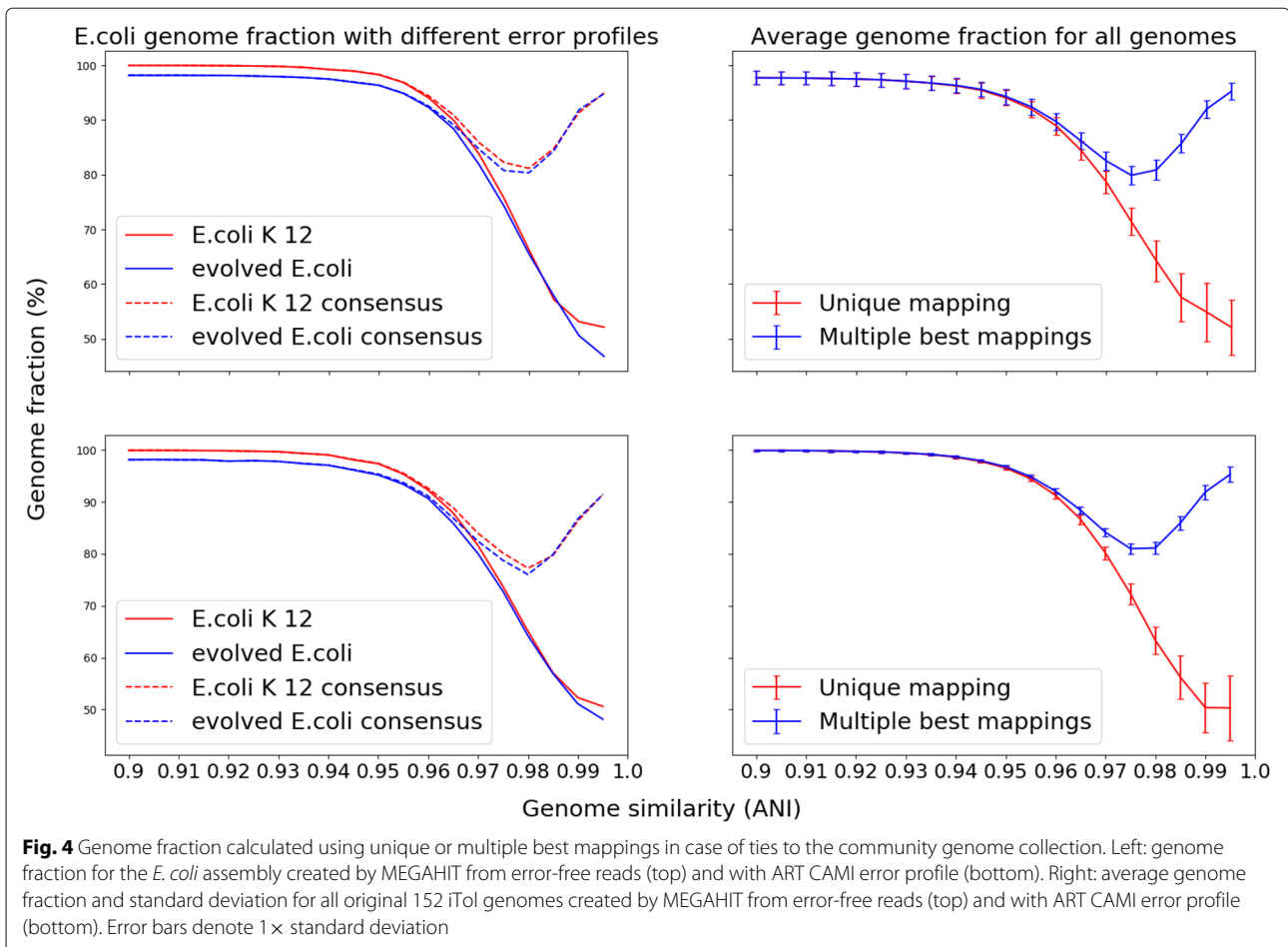
We systematically investigated the effect of related strains on assembler performances across a wide range of taxa and evolutionary divergences, using the genomes of 152 species from the interactive tree of life iTol [48], which includes bacteria, archaea, and eukaryotes. For each genome, we evolved 19 related genomes without larger insertions and deletions and an average nucleotide identity (ANI) between 90% and 99.5% to the original one using steps of 0.5%. For each of the  $152 \cdot 20 = 3040$  pairs of original and evolved genome sequences, we simulated single sample minimal metagenomes at equal genome abundances, with error-free reads at  $50\times$  coverage using wgsim. This constitutes good coverage for the analyzed assemblers, as shown in the previous section. For the resulting samples, variation in assembler performance

should thus primarily be caused by differences in ANI.

The presence of closely related genomes substantially affected assembly quality (Fig. 4). For up to 95% ANI, the assemblers restored high quality assemblies for both genomes. Between 95% and 99% ANI, the genome fraction and assembly size dropped substantially and contig numbers increased. This was the case if we allowed contigs to either map uniquely to one reference genome or to both, in case of multiple optimal mappings. For more closely related genomes, the number of contigs increased drastically and the assembly size continued to drop. The genome fraction remained high when considering non-unique mappings, but decreased for unique mappings; the explanation for this observed behavior is that for an ANI of more than 99%, assemblers produced consensus contigs of the two strains that mostly aligned similarly well to both reference genomes. This was the case for all 152 genomes and their evolved counterparts.

#### Simulating environment-specific data sets

To test the ability to create metagenome data of the human microbiome, we simulated metagenomes from taxonomic profiles of the Human Microbiome Project [9] for different body sites with CAMISIM. We selected 49 samples



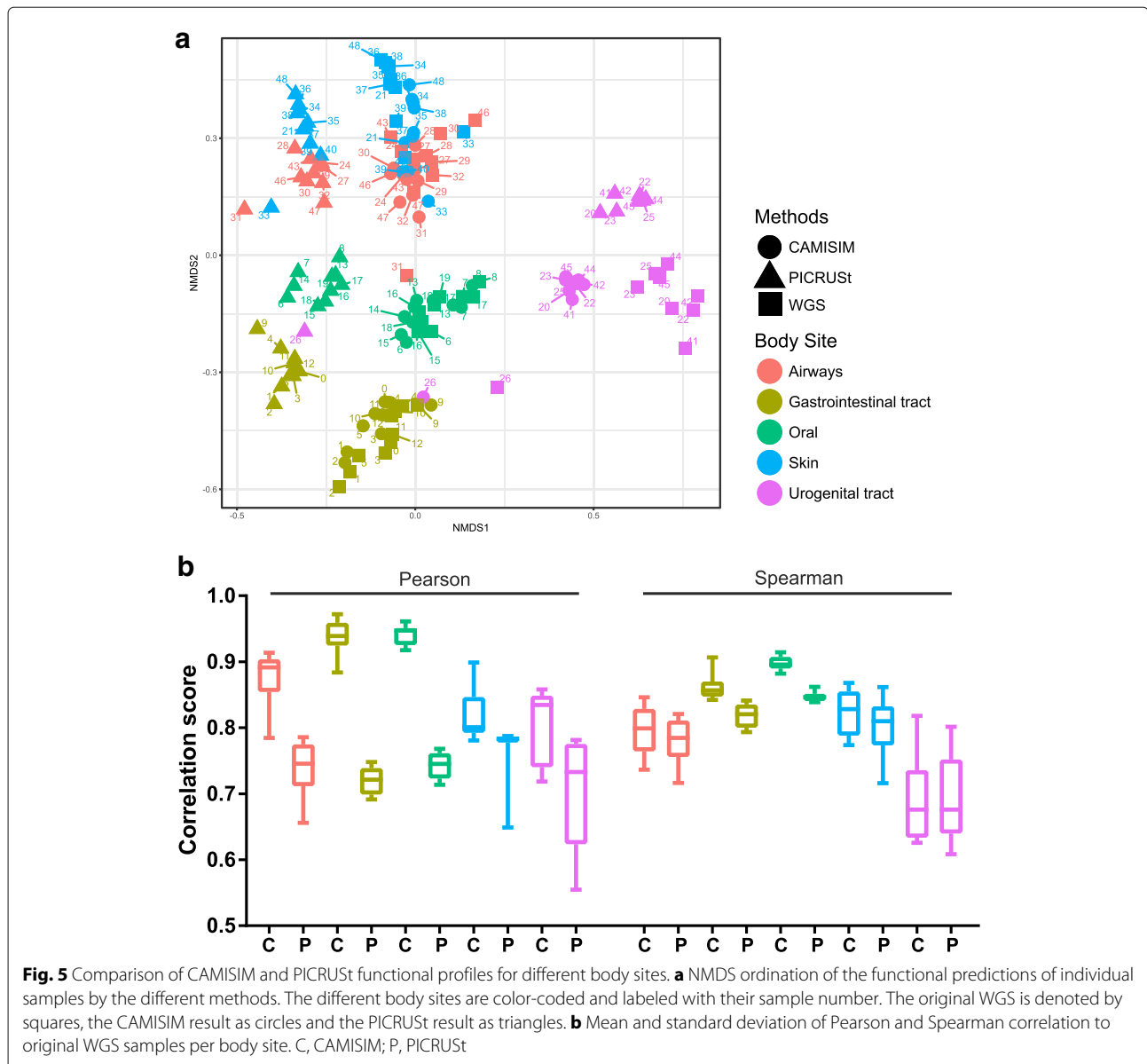
from the airways, gastrointestinal tract, oral cavity, skin and urogenital tract, with whole genome shotgun (WGS) and 16S rRNA gene amplicon sequence data available. We used the published QIIME OTU table (<https://hmpdacc.org/hmp/HMQCP/>) to generate 5 Gb of simulated reads per sample with CAMISIM, resulting in a data set of 245 Gb of Illumina data, and of PacBio data, respectively. Only genomes tagged as “complete genomes” in the NCBI were considered in the data set generation. To decrease the chance of OTUs not being represented by a genome, the option of allowing multiple OTUs being represented by a single reference genome was turned on. This can be relevant for instance when due to sequencing errors in 16S rRNA data, individual community genomes are represented by multiple OTUs.

For a functional comparison of the simulated data with the original metagenome shotgun data, we inferred KEGG Ortholog family abundance profiles from the raw read data sets [49]. To this end, all reads were searched with Diamond v0.9.10 using its blastx command with default options [50] against the KEGG GENES database (release 77, species\_prokaryotes, best-hit approach) and linked to KEGG Orthology (KO) via the KEGG mapping files.

KO profile similarity between the simulated and original metagenome samples was calculated with Pearson’s correlation coefficient (PCC) and Spearman rank correlation (SRC), and visualized with non-metric multidimensional scaling (NMDS) [51].

For comparison, we also created functional profiles with PICRUSt [52], using a prediction model generated from 3772 KEGG genomes and corresponding 16S rRNA gene sequences according to the PICRUSt “Genome Prediction Tutorial” (Additional file 1). The PCC of the CAMISIM and original samples approached a striking 0.97 for body sites with high bacterial abundances and many sequenced genomes available, such as the GI tract and oral cavity, and still ranged from 0.72 to 0.91 for airways, skin and urogenital tract samples (Fig. 5b). All PCCs were 7–30% higher than the PCC of PICRUSt with the original metagenome samples. Thus, CAMISIM created metagenome samples functionally even closer to the original metagenome samples than the functional profiles created by PICRUSt. The higher PCC may also partly be due to the fact that the original and CAMISIM data were annotated by “blasting” reads versus KEGG, while the PICRUSt profiles were directly generated from KEGG





genome annotations. The Spearman correlation of the simulated CAMISIM samples to the original metagenome samples was slightly lower than the PCC across all body sites, and very similar for CAMISIM and PICRUSt (0–6% improvement of CAMISIM over PICRUSt). These results demonstrate the quality of the CAMISIM samples.

The NMDS plot (Fig. 5a) showed a very distinct clustering of the CAMISIM and original WGS samples by body site, more closely than the original samples clustered with the PICRUSt profiles. Even though the urogenital tract samples did not cluster perfectly, the CAMISIM samples still formed a very distinct cluster close to the original one. Even outliers in the original samples

were, at least partly, detected and correctly simulated (both original and simulated sample 26 of urogenital tract cluster most closely with the gastrointestinal tract microbiomes).

We also provide a multi-sample mouse gut data set for software developers to benchmark against. For 64 16S rRNA samples from the mouse gut [17], we simulated 5 Gb of Illumina and PacBio reads each. The mice were obtained from 12 different vendors and the samples characterized by 16S V4 amplicon sequencing (OTU mapping file in Additional file 1). Since for mouse gut only a few complete reference genomes were available, the “scaffold” quality for downloading genomes was chosen.

## Discussion and conclusions

CAMISIM is a flexible program for simulating a large variety of microbial communities and metagenome samples. To our knowledge, it possesses the most complete feature set for simulating realistic microbial communities and metagenome data sets. This feature set includes simulation from taxonomic profiles as templates, inclusion of both natural and simulated strain level diversity, and modelling multi-sample data sets with different underlying community abundance distributions. Read simulators are included for short-read (Illumina) and long-read (PacBio, ONT) sequencing technologies, allowing the generation of hybrid data sets. This turns CAMISIM into a versatile metagenome simulation pipeline, as modules for new (or updated) sequencing technologies and emerging experimental setups can easily be incorporated.

We systematically explored the effect of specific data properties on assembler performances on several thousand minimally challenging metagenomes. While low coverage reduced assembly quality for both assemblers, metaSPAdes and MEGAHIT performed generally well for medium to high coverages and different error profiles. Notably, MEGAHIT is computationally very efficient and overall performed well. As noted before [15, 53], assemblers had problems with resolving closely related genomes in our experiments. For an in-depth investigation, we systematically analyzed the effect of related strains on MEGAHIT's performance across a wide range of taxa and evolutionary divergences. The average nucleotide identity (ANI) between two genomes is a robust measure of genome relatedness; an ANI value of 95% roughly corresponds to a 70% DNA-DNA reassociation value—a historical definition of bacterial species [54, 55]. For an pairwise ANI below 95%, the mixture of strains could be separated quite well and assembled into different contigs. For an ANI of more than 99%, consensus contigs of strains were produced that mostly aligned similarly well to either reference genome. In the “twilight zone” of 95–99% nucleotide identity, assembly performance dropped substantially and MEGAHIT's inability to reliable phase strain variation resulted in many small (and often redundant) contigs. For IDBA-UD [56], another *de Bruijn* graph-based metagenome assembler, a similar pattern has been observed [57], indicating that such behavior is common to many assemblers.

Resolving strains from metagenome shotgun data is an open research question, though recently promising computational approaches were proposed [11, 58]. The hybrid long- and short-read simulated data sets we are providing for mouse gut and human body sites could enable the development of new approaches for this task. CAMISIM will facilitate the generation of further realistic benchmarking data sets to assess their performances. With the advent of long-read metagenomics, metagenomics

software needs to coevolve, e.g., metagenome assemblers should support long-read and hybrid assemblies in the future (metaSPAdes [19] is a pioneer in this regard). In fact, hybrid data sets will be key to the second CAMI challenge [59].

CAMISIM can also be used to study the effect of experimental design (e.g., number of replicates, sequencing depth, insert sizes) or intrinsic community properties, such as taxonomic composition, community abundance distributions, and organismal complexities, on program performance. Due to the enormous diversity of naturally occurring microbial communities, experimental and sequencing technology setups used in the field, such explorations are required to determine the most effective combinations for specific research questions.

While we tried to mimic naturally occurring data sets as close as possible, CAMISIM, especially in the *de novo* mode and when artificially simulating new strains, requires the user to make choices about the underlying evolutionary and ecological parameters. This includes but is not necessarily limited to the organismal abundance distribution and its parameters, like discussed in [29, 30, 32], of microbial communities and the parameters driving strain evolution. When developing metagenome analysis tools, these should not only be entirely optimized to work on individual data sets produced by CAMISIM, but also tested with additional, optimally real world data.

## Availability and requirements

**Project name:** CAMISIM

**Project home page:** <https://github.com/CAMI-challenge/CAMISIM>

**Operating system(s):** UNIX

**Programming language:** Python 2.7

**Other requirements:** <https://github.com/CAMI-challenge/CAMISIM/wiki>

**License:** Apache 2.0

**Any restrictions to use by non-academics:** None.

## Additional file

**Additional file 1:** Supplementary information. Taxonomic profile based community design: BIOM format details, Reference database. *De novo* community design: Creation of the mapping file. Genome assembly metrics. Methods: iTol, Parameters, PICRUST, Configfile. OTU mapping file. (PDF 146 kb)

## Acknowledgements

The authors thank the Isaac Newton Institute for Mathematical Sciences for its hospitality during the programme MTG, which was supported by EPSRC Grant Number EP/K032208/1, and Victoria Sack for generating Fig. 1. This research project has been supported by the President's Initiative and Networking Funds of the Helmholtz Association of German Research Centres (HGF) under contract number VH-GS-202.

### Availability of data and materials

Config files, input genomes, and metadata for the data sets generated and/or analyzed are available on GitHub: <https://github.com/CAMI-challenge/CAMISIM> and <https://github.com/CAMI-challenge/CAMISIM-DATA>.

The large human and mouse gut microbiome data sets (alongside the BIOM and config files from which they were created) are available at: <https://data.cami-challenge.org/participate>.

### Authors' contributions

AF, PH, SM, ED, JD, JF, MZD, AED, and AB implemented CAMISIM. AF and PB tested the software. AF, TRL, and AB performed the experiments. AF, TRL, AS, AB, and ACM interpreted the results. AF, PH, TRL, PB, AB, and ACM wrote the manuscript. AB and ACM conceived the experiments. ACM conceived and supervised the project. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

### Author details

<sup>1</sup>Computational Biology of Infection Research, Helmholtz Centre for Infection Research, 38124 Braunschweig, Germany. <sup>2</sup>Formerly Department of Algorithmic Bioinformatics, Heinrich-Heine University Düsseldorf, 40225 Düsseldorf, Germany. <sup>3</sup>German Center for Infection Research (DZIF), partner site Hannover-Braunschweig, 38124 Braunschweig, Germany. <sup>4</sup>Center for Biotechnology and Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany. <sup>5</sup>The three institute, University of Technology Sydney, Sydney NSW 2007, Australia.

Received: 18 April 2018 Accepted: 21 January 2019

Published online: 08 February 2019

### References

1. Turnbaugh PJ, Ley RE, Hamady M, Fraser-Liggett C, Knight R, Gordon JI. The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*. 2007;449(7164):804–10. <https://doi.org/10.1038/nature06244>.
2. Proctor LM, Sechi S, DiGiacomo ND, Fettweis JM, Jefferson KK, et al. The integrative human microbiome project: dynamic analysis of microbiome-host omics profiles during periods of human health and disease. *Cell Host Microbe*. 2014;16(3):276–89. <https://doi.org/10.1016/j.chom.2014.08.014>.
3. Warnecke F, Luginbühl P, Ivanova N, Ghassemian M, Richardson TH, et al. Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. *Nature*. 2007;450(7169):560–5. <https://doi.org/10.1038/nature06269>.
4. Hess M, Sczyrba A, Egan R, Kim TW, Chokhawala H, et al. Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. *Science*. 2011;331(6016):463–7. <https://doi.org/10.1126/science.1200387>.
5. Bremges A, Maus I, Belmann P, Eikmeyer F, Winkler A, et al. Deeply sequenced metagenome and metatranscriptome of a biogas-producing microbial community from an agricultural production-scale biogas plant. *GigaScience*. 2015;4:33. <https://doi.org/10.1186/s13742-015-0073-6>.
6. Sunagawa S, Coelho LP, Chaffron S, Kultima JR, Labadie K, et al. Ocean plankton. Structure and function of the global ocean microbiome. *Science*. 2015;348(6237):1261–359. <https://doi.org/10.1126/science.1261359>.
7. Xiao L, Feng Q, Liang S, Sonne SB, Xia Z, et al. A catalog of the mouse gut metagenome. *Nat Biotechnol*. 2015;33(10):1103–8. <https://doi.org/10.1038/nbt.3353>.
8. Kunath BJ, Bremges A, Weimann A, McHardy AC, Pope PB. Metagenomics and CAZyme Discovery. *Methods Mol Biol*. 2017;1588:255–77. [https://doi.org/10.1007/978-1-4939-6899-2\\_20](https://doi.org/10.1007/978-1-4939-6899-2_20).
9. Huttenhower C, Gevers D, Knight R, Abubucker S, Badger JH, et al. Structure, function and diversity of the healthy human microbiome. *Nature*. 2012;486(7402):207–14. <https://doi.org/10.1038/nature11234>.
10. Scholz M, Ward DV, Pasolli E, Tolio T, Zolfo M, et al. Strain-level microbial epidemiology and population genomics from shotgun metagenomics. *Nat Methods*. 2016;13(5):435–8. <https://doi.org/10.1038/nmeth.3802>.
11. Quince C, Delmont TO, Raguideau S, Alneberg J, Darling AE, et al. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biol*. 2017;18(1):181. <https://doi.org/10.1186/s13059-017-1309-9>.
12. Thompson LR, Sanders JG, McDonald D, Amir A, Ladau J, et al. A communal catalogue reveals earth's multiscale microbial diversity. *Nature*. 2017. <https://doi.org/10.1038/nature24621>.
13. Quince C, Walker AW, Simpson JT, Loman NJ, Segata N. Shotgun metagenomics, from sampling to analysis. *Nat Biotechnol*. 2017;35(9):833–44. <https://doi.org/10.1038/nbt.3935>.
14. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet*. 2016;17(6):333–51. <https://doi.org/10.1038/nrg.2016.49>.
15. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, et al. Critical assessment of metagenome interpretation—a benchmark of metagenomics software. *Nat Methods*. 2017;14(11):1063–71. <https://doi.org/10.1038/nmeth.4458>.
16. Belmann P, Dröge J, Bremges A, McHardy AC, Sczyrba A, Barton MD. Bioboxes: standardised containers for interchangeable bioinformatics software. *GigaScience*. 2015;4:47. <https://doi.org/10.1186/s13742-015-0087-0>.
17. Roy U, Galvez EJC, Iljazovic A, Lesker TR, Blazewski AJ, et al. Distinct microbial communities trigger colitis development upon intestinal barrier damage via innate or adaptive immune cells. *Cell Rep*. 2017;21(4):994–1008. <https://doi.org/10.1016/j.celrep.2017.09.097>.
18. Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*. 2015;31(10):1674–6. <https://doi.org/10.1093/bioinformatics/btv033>.
19. Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. metaSPAdes: a new versatile metagenomic assembler. *Genome Res*. 2017;27:1959–116. <https://doi.org/10.1101/gr.213959.116>.
20. McDonald D, Clemente JC, Kuczynski J, Rideout JR, Stombaugh J, et al. The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *GigaScience*. 2012;1:7. <https://doi.org/10.1186/2047-217X-1-7>.
21. Pruitt KD, Tatusova T, Maglott DR. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*. 2007;35(Database issue):61–5. <https://doi.org/10.1093/nar/gkl842>.
22. Darling ACE, Mau B, Blattner FR, Perna NT. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*. 2004;14(7):1394–403. <https://doi.org/10.1101/gr.2289704>.
23. Albertsen M, Hugenholtz P, Skarshewski A, Nielsen KL, Tyson GW, Nielsen PH. Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nat Biotechnol*. 2013;31(6):533–8. <https://doi.org/10.1038/nbt.2579>.
24. Bendall ML, Stevens SL, Chan LK, Malfatti S, Schwientek P, et al. Genome-wide selective sweeps and gene-specific sweeps in natural bacterial populations. *The ISME J*. 2016;10(7):1589–601. <https://doi.org/10.1038/ismej.2015.241>.
25. Stolze Y, Bremges A, Rummig M, Henke C, Maus I, et al. Identification and genome reconstruction of abundant distinct taxa in microbiomes from one thermophilic and three mesophilic production-scale biogas plants. *Biotechnol Biofuels*. 2016;9:156. <https://doi.org/10.1186/s13068-016-0565-3>.
26. Roux S, Chan LK, Egan R, Malmstrom RR, McMahon KD, Sullivan MB. Ecogenomics of virophages and their giant virus hosts assessed through time series metagenomics. *Nat Commun*. 2017;8(1). <https://doi.org/10.1038/s41467-017-01086-2>.
27. Alneberg J, Bjarnason BS, de Bruijn I, Schirmer M, Quick J, et al. Binning metagenomic contigs by coverage and composition. *Nat Methods*. 2014;11(11):1144–6. <https://doi.org/10.1038/nmeth.3103>.
28. Kang DD, Froula J, Egan R, Wang Z. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ*. 2015;3:1165. <https://doi.org/10.7717/peerj.1165>.
29. Curtis TP, Sloan WT, Scannell JW. Estimating prokaryotic diversity and its limits. *Proc Natl Acad Sci*. 2002;99(16):10494–9. <https://doi.org/10.1073/pnas.142680199>.
30. Oñifèru ID, Lunn M, Curtis TP, Wells GF, Criddle CS, et al. Combined niche and neutral effects in a microbial wastewater treatment community. *Proc Natl Acad Sci*. 2010;107(35):15345–50. <https://doi.org/10.1073/pnas.100604107>.

31. Ulrich W, Ollik M, Ugland KI. A meta-analysis of species–abundance distributions. *Oikos*. 2010;119(7):1149–55. <https://doi.org/10.1111/j.1600-0706.2009.18236.x>.
32. Unterseher M, Jumpponen A, Opik M, Tedersoo L, Moora M, et al. Species abundance distributions and richness estimations in fungal metagenomics—lessons learned from community ecology. *Mol Ecol*. 2011;20(2):275–85. <https://doi.org/10.1111/j.1365-294X.2010.04948.x>.
33. Nielsen HB, Almeida M, Juncker AS, Rasmussen S, Li J, et al. Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nat Biotechnol*. 2014;32(8):822–8. <https://doi.org/10.1038/nbt.2939>.
34. Imelfort M, Parks D, Woodcroft BJ, Dennis P, Hugenholtz P, Tyson GW. GroopM: an automated tool for the recovery of population genomes from related metagenomes. *PeerJ*. 2014;2:603. <https://doi.org/10.7717/peerj.603>.
35. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. *Bioinformatics*. 2012;28(4):593–4. <https://doi.org/10.1093/bioinformatics/btr708>.
36. Singer E, Andreopoulos B, Bowers RM, Lee J, Deshpande S, et al. Next generation sequencing data of a defined microbial mock community. *Sci Data*. 2016;3:160081. <https://doi.org/10.1038/sdata.2016.81>.
37. Bremges A, Singer E, Woyke T, Sczyrba A, McCorS. Metagenome-enabled error correction of single cell sequencing reads. *Bioinformatics*. 2016;32(14):2199–201. <https://doi.org/10.1093/bioinformatics/btw144>.
38. Singer E, Bushnell B, Coleman-Derr D, Bowman B, Bowers RM, et al. High-resolution phylogenetic microbial community profiling. *ISME J*. 2016;10(8):2020–032. <https://doi.org/10.1038/ismej.2015.249>.
39. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*. 2009;25(16):2078–9. <https://doi.org/10.1093/bioinformatics/btp352>.
40. Ono Y, Asai K, Hamada M. PBSIM: PacBio reads simulator—toward accurate genome assembly. *Bioinformatics*. 2013;29(1):119–21. <https://doi.org/10.1093/bioinformatics/bts649>.
41. Yang C, Chu J, Warren RL, Birol I. NanoSim: nanopore sequence read simulator based on statistical characterization. *GigaScience*. 2017. <https://doi.org/10.1093/gigascience/gix010>.
42. Jia B, Xuan L, Cai K, Hu Z, Ma L, Wei C. NeSSM: a next-generation sequencing simulator for metagenomics. *PLoS ONE*. 2013;8(10):75448. <https://doi.org/10.1371/journal.pone.0075448>.
43. Johnson S, Trost B, Long JR, Pittet V, Kusalik A. A better sequence-read simulator program for metagenomics. *BMC Bioinformatics*. 2014;15(Suppl 9):14. <https://doi.org/10.1186/1471-2105-15-s9-s14>.
44. Shcherbina A. FASTQSim: platform-independent data characterization and in silico read generation for NGS datasets. *BMC Res Notes*. 2014;7(1):533. <https://doi.org/10.1186/1756-0500-7-533>.
45. Angly FE, Willner D, Rohwer F, Hugenholtz P, Tyson GW. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Res*. 2012;40(12):94–4. <https://doi.org/10.1093/nar/gks251>.
46. Rinke C, Schwientek P, Sczyrba A, Ivanova NN, Anderson IJ, et al. Insights into the phylogeny and coding potential of microbial dark matter. *Nature*. 2013;499(7459):431–7. <https://doi.org/10.1038/nature12352>.
47. Laehnemann D, Borkhardt A, McHardy AC. Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction. *Brief Bioinformatics*. 2016;17(1):154–79. <https://doi.org/10.1093/bib/bbv029>.
48. Letunic I, Bork P. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. *Bioinformatics*. 2007;23(1):127–8. <https://doi.org/10.1093/bioinformatics/btl529>.
49. Kanehisa M, Sato Y, Kawashima M, Furumichi M, Tanabe M. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res*. 2015;44(D1):457–62. <https://doi.org/10.1093/nar/gkv1070>.
50. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. *Nat Methods*. 2014;12(1):59–60. <https://doi.org/10.1038/nmeth.3176>.
51. Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*. 1964;29(1):1–27. <https://doi.org/10.1007/bf02289565>.
52. Langille MGI, Zaneveld J, Caporaso JG, McDonald D, Knights D, et al. Predictive functional profiling of microbial communities using 16s rna marker gene sequences. *Nat Biotech*. 2013;31(9):814–21. <https://doi.org/10.1038/nbt.2676>.
53. Awad S, Irber L, Brown CT. Evaluating metagenome assembly on a simple defined community with many strain variants. *bioRxiv*. 2017. <https://doi.org/10.1101/155358>.
54. Konstantinidis KT, Tiedje JM. Genomic insights that advance the species definition for prokaryotes. *Proc Natl Acad Sci USA*. 2005. <https://doi.org/10.1073/pnas.0409727102>.
55. Varghese NJ, Mukherjee S, Ivanova N, Konstantinidis KT, Mavrommatis K, et al. Microbial species delineation using whole genome sequences. *Nucleic Acids Res*. 2015. <https://doi.org/10.1093/nar/gkv657>.
56. Peng Y, Leung HCM, Yiu SM, Chin FYL. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*. 2012;28(11):1420–8. <https://doi.org/10.1093/bioinformatics/bts174>.
57. DeMaere MZ, Darling AE. Deconvoluting simulated metagenomes: the performance of hard- and soft- clustering algorithms applied to metagenomic chromosome conformation capture (3c). *PeerJ*. 2016;4:2676. <https://doi.org/10.7717/peerj.2676>.
58. Cleary B, Brito IL, Huang K, Gevers D, Shea T, et al. Detection of low-abundance bacterial strains in metagenomic datasets by eigengenome partitioning. *Nat Biotechnol*. 2015;33(10):1053–60. <https://doi.org/10.1038/nbt.3329>.
59. Bremges A, McHardy AC. Critical Assessment of Metagenome Interpretation Enters the Second Round. *mSystems*. 2018;3(4). <https://doi.org/10.1128/mSystems.00103-18>.
60. Wick RR, Schultz MB, Zobel J, Holt KE. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*. 2015;31(20):3350–2. <https://doi.org/10.1093/bioinformatics/btv383>.
61. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. *Bioinformatics*. 2013;29(8):1072–5. <https://doi.org/10.1093/bioinformatics/btt086>.
62. Richter DC, Ott F, Auch AF, Schmid R, Huson DH. MetaSim — a sequencing simulator for genomics and metagenomics. *PLoS ONE*. 2008;3(10):3373. <https://doi.org/10.1371/journal.pone.0003373>.
63. Mende DR, Waller AS, Sunagawa S, Järvelin AL, Chan MM, et al. Assessment of metagenomic assembly using simulated next generation sequencing data. *PLoS ONE*. 2012;7(2):31386. <https://doi.org/10.1371/journal.pone.0031386>.
64. Bushnell B. BBMap: A fast, accurate, splice-aware aligner; 2014. <https://sourceforge.net/projects/bbmap>. Accessed 30 Jan 2019.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

